**Amendments to the Claims:**

1. (currently amended) A ~~Method~~ method to secure the execution of a program in an electronic assembly ~~comprising~~ having information processing means and information storage means, ~~characterised in that it consists in~~ the method comprising:

- checking the execution time of at least one sequence of said program with respect to the normal predetermined execution time of said sequence, by:

- planning an end of said normal predetermined execution time;

- planning a point of arrival of said at least one sequence of said program according to the normal predetermined execution time of said sequence;

- starting a counter timer with associated interrupt at the point of departure of execution of said sequence;

- delivering an interrupt on expiry of said counter timer, wherein the expiry of said counter timer corresponds to the planned end of the normal predetermined execution time of said sequence;

- determining an actual point of arrival of said sequence when said interrupt is delivered;

- checking if the determined actual point of arrival of said sequence corresponds to said planned point of arrival.

2. (cancelled)

3. (cancelled)

4. (cancelled)

5. (currently amended) The ~~Method~~ method according to claim 1~~one of claims 1 to 4~~, ~~characterised in that it consists in~~ further comprising:

- checking that the planned point of arrival of said sequence is reached after completion of the normal predetermined execution time period ~~the execution time of at least one sequence of said program with respect to the normal predetermined~~

~~execution time of said sequence~~ so as to protect against attacks disturbing the execution of said program.

6. (currently amended) The ~~Method~~ method according to ~~one of~~ claim~~s~~ 1 ~~to 5~~, ~~characterised in that it consists in~~ further comprising:

- triggering at the start of said sequence an interrupt counter initialised to the value of the normal predetermined execution time of said sequence, by:

- triggering an interrupt in the program execution on expiry of the counter; and

- diverting execution of said program to an interrupt management routine in order to check the point of arrival of said sequence.

7. (currently amended) The ~~Method~~ method according to ~~one of the previous~~ claim~~s~~ 6, ~~characterised in that~~ comprising:

- if the execution time of said sequence is not normal, said ~~the~~ interrupt management routine is immediately followed by a sequence to set a fraud indicator in memory or by an interruption of the current execution by another means.

8. (currently amended) The ~~Method~~ method according to ~~one of the previous~~ claim~~s~~ 1, ~~characterised in that it consists in~~ further comprising:

- adding to said sequence instructions or loops or equivalent so as to equalise the execution time of the sequence in all its branches or so that the execution time of said sequence is modified if there is an attack.

9. (currently amended) The ~~Method~~ method according to claim 6, ~~characterised in that~~ wherein the interrupt management routine is placed at the last location of the program memory or just before a shared domain boundary so as to leave the permitted program memory area if an attack prevents execution of the interrupt return.

10. (currently amended) An ~~Electronic~~ electronic module ~~comprising~~ having information processing means and information storage means containing a program to be executed, ~~characterised in that it~~ the electronic module ~~comprises~~ comprising:

- checking means including a counter timer with triggering of an interrupt on expiry to check the execution time of at least one sequence of said program with respect to the normal predetermined execution time of said sequence, wherein:

- said normal predetermined execution time being determined on expiry of the counter timer; and

- said checking means being arranged for triggering said counter timer at the point of departure of execution of said at least one sequence of said program.

11. (currently amended) The ~~Electronic~~ electronic ~~Module~~ module according to claim 10, ~~characterised in that~~ wherein the checking means ~~comprise~~ includes means for diverting execution of said program to an interrupt management routine in order to check the point of arrival of said sequence ~~a counter with triggering of an interrupt on expiry~~.

12. (currently amended) A ~~Card~~ card ~~characterised in that it~~ comprising the electronic module according to claim 10 or 11.

13. (currently amended) A ~~Computer~~ computer program including program code instructions to execute steps of the method according to ~~one of~~ claim~~s~~ 1 ~~to 9~~ when said program is run in a computer system.

14. (new) The method according to claim 5, comprising:

- triggering at the start of said sequence an interrupt counter initialised to the value of the normal predetermined execution time of said sequence,

- triggering an interrupt in the program execution on expiry of the counter, and

- diverting execution of said program to an interrupt management routine in order to check the point of arrival of said sequence.

15. (new) The method according to claim 14, wherein if the execution time of said sequence is not normal, said interrupt management routine is immediately followed by a sequence to set a fraud indicator in memory or by an interruption of the current execution by another means.

16. (new) The method according to claim 5, further comprising:

- adding to said sequence instructions or loops or equivalent so as to equalise the execution time of the sequence in all its branches or so that the execution time of said sequence is modified if there is an attack.

17. (new) The method according to claim 6, further comprising:

- adding to said sequence instructions or loops or equivalent so as to equalise the execution time of the sequence in all its branches or so that the execution time of said sequence is modified if there is an attack.

18. (new) The method according to claim 7, further comprising:

- adding to said sequence instructions or loops or equivalent so as to equalise the execution time of the sequence in all its branches or so that the execution time of said sequence is modified if there is an attack.